Tero Kauria

Robotin ja konenäköjärjestelmän liittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinöörityö

2.5.2013

Metropolia

Tekijä(t) Otsikko Sivumäärä Aika	Tero Kauria Robotin ja konenäköjärjestelmän liittäminen 22 sivua + 1 liitettä 2.5.2013
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Prosessiautomaatio
Ohjaaja(t)	Field Application Engineer Petri Pitkälä Yliopettaja Jouni Jokelainen

Työ tehtiin Omron Electronics Oy:lle opetus- ja asiakastarpeisiin. Yamaha-Omron SCARA -robotti oli ensimmäinen Suomeen tullut kappale kyseisestä mallista eikä ollut kokemusta Suomessa. Päämääränä oli tuottaa malliohjelma, jota voidaan käyttää asiakkaiden koulutukseen ja esimerkkinä asiakkaille siitä, miten kappaleiden lavausohjelma voidaan tehdä.

Tärkeänä elementtinä työssä oli saada kommunikointi toimimaan konenäköjärjestelmän ja SCARA-robotin välillä ilman ylimääräistä logiikkaa ja täten saavuttaa kompakti ja kustannustehokas lavausrobotti, joka pystyy ottamaan kappaleita loogisessa järjestyksessä kuljetinhihnalta pysäyttämättä kuljetinta.

Olennaisena osana näiden päämäärien saavuttamiseen on tietää teoria SCARA-robottien, kommunikointien ja konenäköjärjestelmän taustalla, joten työn aikana perehdytään servotekniikkaan, paikka-antureihin, tiedonsiirtoon ja konenäköjärjestelmän teoriaan.

Työssä ilmeni ongelma kommunikoinnissa, jolloin SACRA-robotti ja konenäköjärjestelmä jäivät lähettämään toisilleen virhesanomaa. Tämä näkyi verkossa kasvaneena liikennemääränä, mutta ei vaikuttanut laitteen toimintaan.

Lopputuloksena oli onnistunut projekti, jossa tapahtui virheitä ja niistä opittiin. Kyseistä ohjelmaa on käytetty koulutustarpeissa ja asiakasesittelyissä.

Avainsanat	SCARA-robotti, konenäköjärjestelmä, kommunikointi

Author(s) Title Number of Pages Date	Tero Kauria Connecting robot and vision system 22 pages + 1 appendices 21 April 2013
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Process automation
Instructor(s)	Petri Pitkälä, Field Application Engineer Jouni Jokelainen, Senior Lecturer

This thesis was made to Omron Electronics Oy for teaching and demonstration purposes. Omron-Yamaha SCARA-robot was first one that arrived in Finland and there was no experience about this model before this work. Main purpose of this work was to demonstrate capabilities of this robot and make example how it can been used.

One of the main points was to make communication between SCARA-robot and vision system. Communication was made without PLC. TCP/IP-communication was established between SCARA and vision system.

At work there was few errors in communication, but they didn't affect communication between TCP/IP-server and client. Only way to detect this error was with Wireshark capture from network.

Result of this project was compact solution to small palletizing cells, where is need to pick objects from product stream.

Keywords	SCARA-robot, Vision system, Communication

Sisällys

1 Johdanto 1

2 Robotti 2

2.1 Servo 2
2.2 Paikka-anturi 4
2.2.1 Inkrementtianturi 5
2.2.2 Resolveri-paikka-anturit 6
2.2.3 Absoluutti-paikka-anturi 7
2.3 SCARA-robotti 8
2.4 Ohjain 9

3 Konenäkö 10

3.1 Valaistus 11
3.2 Kamera 11
3.3 Suodattimet 12

4 Ethernet-liikenne 13

4.1 IP-osoite 13
4.2 QOS-palvelu 14
4.3 TCP/IP-kommunikointi 14
4.4 UDP-kommunikointi 16

5 Robotin ohjelma 17

5.1 Konenäön ja robotin välinen kommunikointi 17
5.2 Liikkuvan kohteen seuraaminen 18
5.3 Pinoaminen 19

6 Konenäön ohjelma 21

7 Loppusanat 22

Lähteet 23

Liitteet

Liite 1. SCARA-robotin ohjelma

Lyhenteet

DeviceNet	Teollisuustiedonsiirtostandardi. Tarjoaa varmemman tiedonsiirron verrattuna RS-485-kommunikointiin.
EIA-485	Tunnetaan myös RS-485-kommunikointina, käyttää kahta johdinta tiedon siirtämiseen.
IP	Internet Protocol. Määrittelee perusteet Ethernet-kommunikoinnille.
RVDT	Rotary Variable Differential Transformer. Anturi, jolla saadaan akselin asento.
SCARA	Selective Compliant Assembly Robot Arm. SCARA-robotit jäljittelevät ihmisen kättä saavuttaakseen erinomaisen nopeuden.
Servo	Pyörivä moottori, jolla saavutetaan erinomainen tarkkuus takaisinkytkentä anturin avulla.
TCP	Transmission Control Protocol. TCP määrittelee miten vikatilanteissa tulee toimia ja tarjoaa protokollan, jolla voidaan kommunikoida laitteiden välillä.
UDP	User Datagram Protocol. TCP-kommunikointiin verrattuna UDP on pelkistetty ja eikä sisällä määrittelyä vikatilanteiden hallintaan.
TCP/IP	TCP/IP:lla tarkoitetaan TCP-kommunikointia Ethernet-verkon ylitse useamman laitteen välillä.
UDP/IP	UDP/IP:lla tarkoitetaan UDP-kommunikointia Ethernet-verkon ylitse useamman laitteen välillä. Ei tarjoa vikatilanteiden hallintaa.
STRING	String on muuttujan tyyppimäärittely. String-muuttujaan voidaan kirjoittaa ymmärrettävää tekstiä.
QOS	Quality Of Service. Tällä voidaan määritellä viestin tärkeys, jolla viesti kulkee Ethernet-verkon läpi.

1 Johdanto

Työ tehdään Omron Electronics Oy:lle. Omron on perustettu vuonna 1933 Japanissa. Nykyään Omron työllistää yli 37 000 henkeä, yli 36 eri maassa. Työssä käytetään Omron-Yamaha SCARA -robottia. Kyseistä sarjaa ei ole aikaisemmin käytetty Suomessa, joten kokemusta tästä ei ole. SCARA-robotti on tarkoitus liittää kommunikoimaan konenäköjärjestelmän kanssa ilman ylimääräistä logiikkaa, minkä seurauksena saavutetaan kustannustehokas ratkaisu lavaussovelluksiin.

Työssä on tarkoituksena perehtyä siihen, miten Omron-Yamaha SCARA -robotit toimivat, SCARAn kannalta oleellisten komponenttien toimintaperiaatteisiin ja käyttötarkoituksiin. SCARA-robotti on tarkoitus liittää TCP/IP-kommunikoinnilla Omronin Xpectia-konenäköjärjestelmään ja tutkia, miten TCP/IP-kommunikointi luodaan kahden laitteen välille. Lisäksi on tarkoitus tarkkailla tätä liikennettä ja todentaa sen toimimen.

Työssä on myös tarkoitus testata, miten saadaan luotettava ja tarkka robottiohjelma, jolla nostetaan kappaleita tuotevirrasta ja laitetaan nämä kappaleet toiselle kuljettimelle siistiin muodostelmaan. Kuljettimella voitaisiin kuvata esimerkiksi kuormalavaa, jolle kappaleet pinotaan valmiiksi kuljetettavaksi. Konenäköjärjestelmään luodaan ohjelma, jolla saadaan otettua kuva, josta saadaan kappaleiden X- ja Y-paikat sekä R-kulma ja lähetettyä ne SCARA-robotin käytettäväksi.

Työn tärkeimpänä elementtinä on luoda tästä projektista hyvä malliesimerkki, jolla voidaan näyttää asiakkaille laitteiden toimintaa ja joka toimii myös tämän kaltaisten projektien mallipohjana, josta pienellä muokkauksilla saadaan helposti hyödynnettävä projekti. Projektin lopuksi annetaan palautetta Omronin tuotekehitykselle, jolloin saadaan kehitettyä laitteita käyttäjäystävällisemmäksi ja monipuolisemmiksi.

2 Robotti

Robotteja käytetään teollisuudessa nykyään yhä enemmän, johtuen niiden ominaisuuksista. Robotit voivat tehdä samaa työtä ympäri vuorokauden nopeammin ja tarkemmin kuin ihminen. Robottien yleistymistä hidastaa niiden kallis alkuinvestointi. Robottien tarkkuus saavutetaan servo-moottoreilla ja niihin liitettävistä takaisinkytkentäantureiden avulla. Roboteilla voidaan myös nostaa raskaita taakkoja, joita ihminen ei ilman apuvälineitä pysty nostamaan.

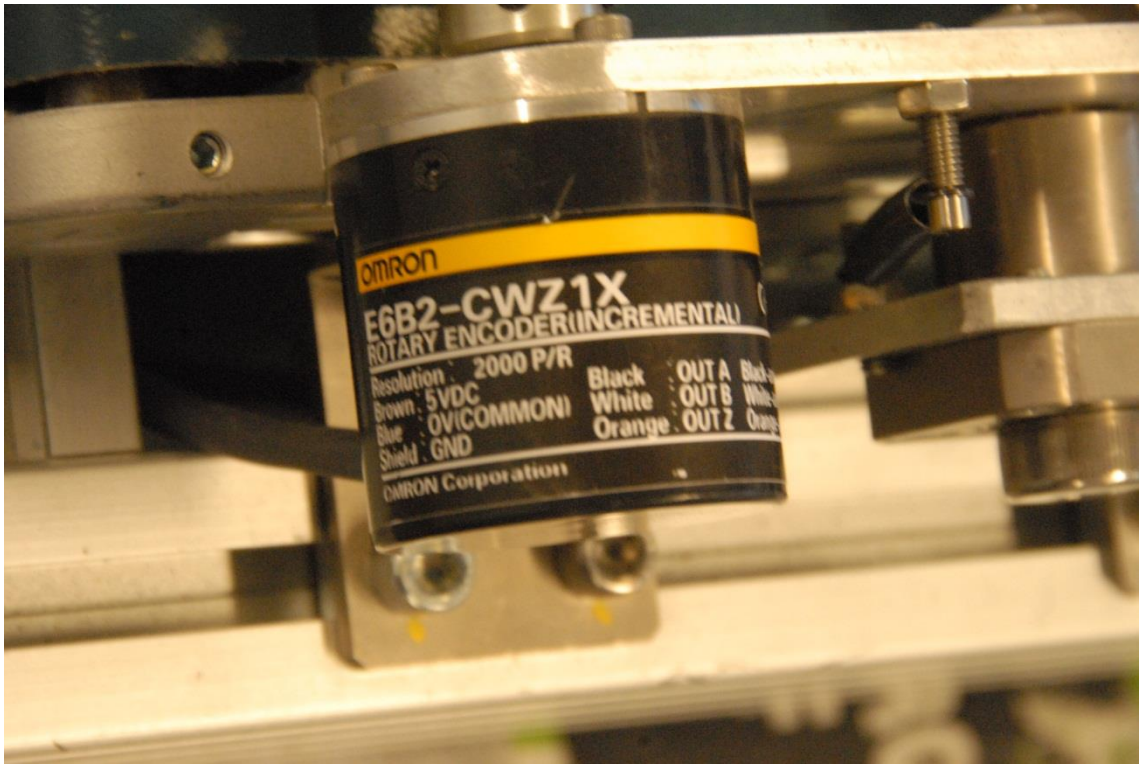
2.1 Servo

Servo-moottorit (kuva 1) eroavat perinteisistä oikosulkumoottoreista usealla tavalla. Perinteisiä oikosulkumoottoreita voisi kuvailla norsuiksi ja servomoottoreita gepardeiksi. Perinteiset oikosulkumoottorit ovat halpoja, kestäviä ja helppoja ohjata. Toisaalta niillä on vaikeaa saada tehtyä tarkkaa paikoitusajoa, ne ovat isoja, painavia sekä kiihtyvät ja pysähtyvät hitaasti. Servomoottorit ovat taas nopeita kiihtymään ja pysähtymään sekä tarkkoja, pieniä ja kevyitä. Huonoina puolina servomoottoreissa ovat monimutkaisempi rakenne, ne tarvitsevat takaisinkytkennän moottorista sekä ovat kalliita valmistaa [1].

Kuva1. Servomoottorin läpileikkaus.[1]

Servo-järjestelmät muodostuvat useasta osasta. Servomoottoriin on liitettynä paikka-anturi, jolla saadaan tiedettyä servomoottorin tarkka asento. Takaisinkytkentä johdotetaan servo-vahvistimelle, joka ohjaa servomoottoria määrättyä kohdetta kohti. Servovahvistimen sisällä lasketaan kohteen ja nykyisen sijainnin erotusta. Sen avulla määrätään uusi voima-arvo servomoottorille, jolla moottori pyrkii kohti kohdetta [2].

Servomoottoreihin liitettäviä takaisinkytkentäantureita (kuva 2) on kahta eri tyyppiä: inkrementti- ja absoluutti-malleja. Absoluutti-mallinen paikka-anturi muistaa oman sijaintinsa, vaikka laitteesta otettaisiin sähköt pois. Tällöin ei tarvitse sähköjen kytkemisen jälkeen ajaa kotiinajoa, joka on pakko tehdä inkrementti-anturilla, jos halutaan ajaa ennalta määrättyyn paikaan. Mikäli inkrementti-anturilla varustettua servomoottorilla ei ajeta kotiinajoa ennen varsinaista ajoa, ei servomoottorin akselin tarkkaa sijaintia tiedetä, eikä voida ajaa absoluuttipaikoitusajoa, vaan pelkästään inkrementaalista ajoa [1;2].



Kuva 2. Paikka-anturi.

Servovahvistimet suunnitellaan ennalta määriteltyjä servomoottoreita varten, jolloin saadaan paras mahdollinen tarkkuus. Tämä eroaa huomattavasti taajuusmuuttajien suunnittelusta. Taajuusmuuttajissa ainoana rajoituksena sille, minkälainen oikosulkumoottori

siihen voidaan liittää, on sen tehoarvo. Servomoottoreita myydään tietyntyyppisen servovahvistimen kanssa. Servovahvistimilla voidaan servoa ajaa kolmella eri tehtävän tyylillä: paikoitus-, nopeus- ja momenttiajolla. [2;3]

Paikoitusajossa servomoottorille määrätään uusi voima, jolla se pyrkii eteenpäin määrättyä kohdetta kohti. Tämä saadaan laskettua encoder-anturilta satavan sijainnin ja halutun kohteen erotuksesta. Eri servojärjestelmävalmistajilla on suuri ero laskentakaavoissa. Osa valmistajista ottaa huomioon kuorman määrän tai muutoksen laskennassa, jolloin saadaan parempi paikoitustarkkuus muutoksesta huolimatta. Mikäli kuorman määrää ei huomioida, tulee esimerkiksi nopeudessa virhettä, joka taas saattaa aiheuttaa virhettä paikoituksessa tai seurantavirheenä interpolaatioajoissa**.** Seurantavirhe lasketaan käsketyn ja nykyisen paikan erotuksena. [2]

Nopeusajossa laskenta on hieman erilaista. Siinä lasketaan nykyisen nopeuden ja käsketyn nopeuden erotus, josta saadaan laskettua uusi voima, jolla pidetään todellinen nopeus käsketyssä nopeudessa. [2;3]

Momenttiajossa seurataan servomoottorin käyttämää tehoa, jolloin tiedetään, kuinka paljon servomoottori tekee työtä, joka on taas verrannollinen momenttiin, jolla servomoottori vääntää. [2;3]

Työssä käytetyssä SCARA-robotissa on absoluuttimalliset servomoottorit. Encoder-anturien tarkkuus on 16 384 pulssia kierroksella. Tällä saadaan noin 0,022 astetta/ pulssi. Teoreettinen tarkkuus akselia kohden on siis noin 0,022 astetta. Lopulliseen tarkkuuteen vaikuttavat lisäksi vaihteiston välykset, hihnavetojen välykset ja muut mekaaniset ominaisuudet. SCARA-robotteja käytetään paikoitustehtävissä, joissa ajetaan johonkin ennalta määrättyyn pisteeseen XYZ-koordinaatistossa.

2.2 Paikka-anturi

Paikka-anturit ovat servokäytöissä kriittisessä osassa sovelluksen kannalta. Anturin ominaisuudet vaikuttavat sovelluksen käyttöominaisuuksiin. Erilaisia paikka-antureita ovat muun muassa seuraavat: resolveri-, inkrementti-, absoluutti- ja RVDT-paikka-anturit. Osa paikka-antureista on tarkempia, osalla on taas paremmat ympäristön olosuhteisiin sopivat ominaisuudet. [2]

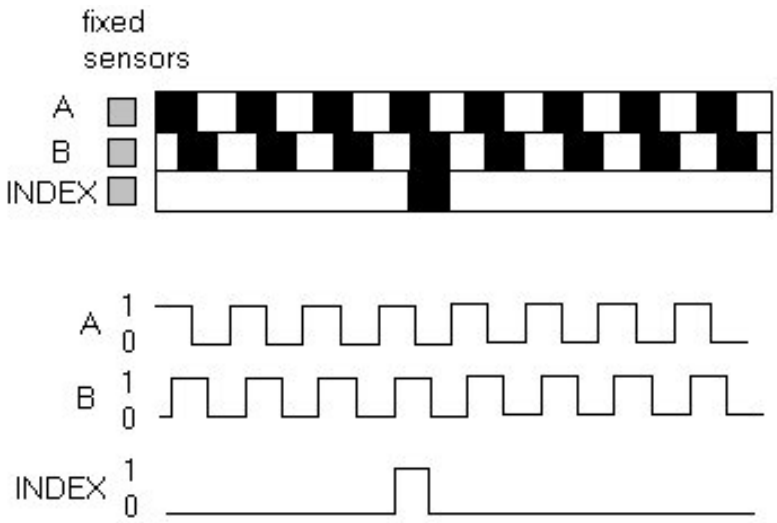
Tästä hyvänä esimerkkinä ovat optiset inkrementti-anturit, joiden hyvinä ominaisuuksina ovat halpa hinta ja hyvä ympäristön elektronisen häiriön sietokyky. Huonoina puolina ovat niiden huono tärinän sieto johtuen rakenteesta, ja mikäli kotelon sisälle pääsee pölyä, aiheutuu tästä ongelmia optisessa mittauksessa. [2]

Robotti-sovelluksiin on taas suotavaa laittaa absoluutti-malliset paikka-anturit, koska nämä anturit tietävät sijaintinsa silloinkin, kun laitteistosta on otettu sähköt pois, jolloin robottia ei tarvitse ajaa kotiasemaan sähkökatkon jälkeen. Tällöin myös robotin tarkkuus paranee, koska kotiinajossa saattaa tulla muutaman millin heitto. Tästä aiheutuisi sovelluksessa pientä epätarkkuutta [3].

Tässä työssä SCARA-robotissa on jokaisella akselilla absoluutti-malliset anturit ja kuljetinhihnalla, josta kappaleista otetaan kuvat, inkrementti-mallinen paikka-anturi. Kuljettimeen ei ole järkevää sijoittaa muuta anturimallia, koska inkrementtianturi on halpa, tarpeeksi hyvä ja tarpeeksi tarkka tähän tehtävään.

2.2.1 Inkrementti-anturi

Inkrementti-paikka-anturi tuottaa akselin pyöriessä pulsseja, joita lasketaan vahvistimella paikkatiedoksi. Kyseiset optiset paikka-anturit perustuvat Moirén periaatteeseen. Anturin sisällä on pyörivä kiekko, joka on kiinnitetty pyörivään akseliin. Kiekkoon on valmistusvaiheessa laitettu tiheästi viivoja, joiden läpi valo heijastetaan. Tällöin vastaanottimen puolella nähdään akselin pyöriminen ja suunta (kuva 3). Tämän jälkeen anturin piirillä muodostetaan pulssit A ja B. Näiden lisäksi yleensä inkrementti-antureissa on Z-pulssi, joka saadaan kerran kierroksessa. Tällöin pystytään havaitsemaan, kun anturi on pyörähtänyt kierroksen. A- ja B-signaalien välinen ero on 90 astetta, ja sen seurauksena pystytään päättelemään, kumpaan suuntaan anturi pyörii. [2;4]



Kuva 3. Inkrementaali-paikka-anturin toimintaperiaate.

Kiekkoon voidaan nykymenetelmillä tehdä 4096 viivaa, jolloin saadaan anturin resoluutioksi 16384, jolla saadaan teoreettiseksi tarkkuudeksi 0,022 astetta. Neljällä kertominen johtuu siitä, että A-pulssilla on nouseva ja laskeva reuna, huomataan näissä kohdissa, että anturi on liikkunut, ja kun lisätään B-pulssien nousevat ja laskevat, saadaan yhteensä kertoimeksi 4 [2].

Anturin hyvänä puolena on sen elektromagneettisten häiriöiden sietokyky. Tämä ominaisuus on tärkeä, koska sähköiset moottorit ja servo vahvistimet tuottavat ympäristöönsä elektromagneettisia häiriöitä. Yksinkertainen rakenne taas tekee siitä halvan paikka-anturin. Huonona puolena anturilla on sen huono pölyn ja tärinän sietokyky. Pölyn joutuessa anturin sisälle saattaa tämä aiheuttaa ongelmia optisessa mittauksessa, ja tärinä saattaa naarmuttaa kiekkoa, jolloin anturi on käyttökelvoton [2].

2.2.2 Resolveri-paikka-anturit

Resolveri-anturit ovat analogisia antureita. Anturissa on sisään rakennettu kaksi kelaa ja akselilla on yksi kela. Akselilla olevaan kelaan syötetään referenssisignaalia, joka on tyypillisesti noin 6-8 kHz taajuista sini-signaalia. Rungossa olevat kelat on sijoitettu 90 asteen välein, jolloin niistä muodostuu sin- ja cos-takaisinkytkentäsignaalit. Näin ollen akselin pyöriessä akselin kelasta muodostuva magneettikenttä vaikuttaa eri tavalla rungossa oleviin keloihin, ja näistä voidaan päätellä akselin asento. [2]

Anturin huonona puolena on sen huono magneettisen häiriön sietokyky, johtuen sen toimintaperiaatteesta. Hyvänä puolena taas voidaan pitää sen tarkkuutta. Analogia-signaalilla on ääretön määrä arvoja, jolloin sen ainoaksi rajoittavaksi tekijäksi muodostuu analogia/digitaali-muuntimen resoluutio. Anturin toimintaperiaatteesta johtuen se ei ole arka tärinälle eikä pölylle. [2]

2.2.3 Absoluutti-paikka-anturi

Absoluutti-malliset paikka-anturit tietävät asentonsa akselilla ilman, että tarvitsi tehdä kotiinajoa sähköjen tultua päälle. Tämä ominaisuus on erinomainen robottisovelluksille. Tällöin robotti voidaan ajaa kotiasemaan sähkön kytkeydyttyä päälle turvallisesti. Absoluutti-anturin akselilla olevaan kiekkoon on käytetty joko Gray-koodia (kuva 4) tai viivakoodia. Gray-koodissa olevat viivat ovat uniikkeja jokaisessa akselin kohdassa, jolloin tiedetään sen sijainti sähköjen kytkeydyttyä. Tähän samaan ideaan perustuvat binäärikoodeilla tehdyt kiekot. [2;5]

Kuva 4. Gray-koodiin perustuva kiekko. [5]

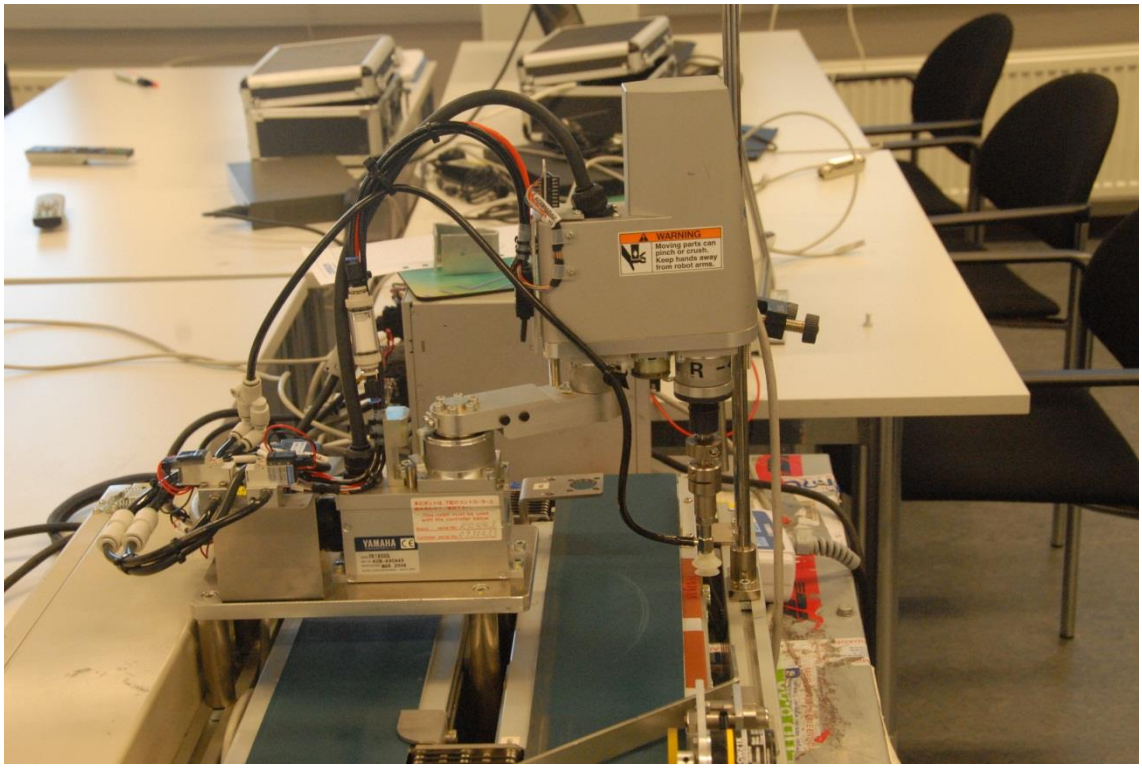
Jotta absoluutti-anturi tietäisi paikkansa sähkökatkojen aikana, on ne yleensä varustettu patterilla. Tästä aiheutuu kunnossapidon kannalta yksi tärkeä huoltokohde. Mikäli anturista loppuu patteri, anturi unohtaa, montako kierrosta se on kotipaikastaan. Anturi kyllä tietää oman sijaintinsa kierroksella, mutta se täytyy ajaa kotipaikkaan ja nollata uudestaan [2].

Inkrementalianturin A-, B- ja Z-pulssi kommunikointia ei voida käyttää absoluutti-anturilla, koska absoluutti-anturin pitää lähettää sen absoluuttinen paikka. Tästä johtuen anturit kommunikoivat vahvistimen kanssa tyypillisesti fyysisen EIA-485 kommunikointitason kautta. Vahvistin kysyy anturilta sijaintia, jolloin anturi lähettää takaisin vastauksen, jossa on paikkatieto. Kehittyneimmille antureille voidaan antaa kotipisteestä poikkeava

kotipiste, jolloin voidaan koneen käyttöönotossa määritellä anturin kotisijainti erittäin tarkaksi. Absoluutti-anturit ovat nykypäivänä yhtä tarkkoja kuin inkrementaali-anturit. [2]

2.3 SCARA-robotti

SCARA-robotteja käytetään pääsääntöisesti kokoonpano-, kappaleiden siirtely-, liimaus- ja muissa samantyylisissä tehtävissä. SCARA-roboteilla (kuva 5) on tyypillisesti 4 vapausastetta.



Kuva 5. SCARA-robotti.

SCARA-robotin kanssa samantyylisiä tehtäviä tekevät karteesiset robotit (kuva 6). SCARA-robotin etuna on karteesisen robotin pienempi asennuspinta-ala. SCARA-robotilla on yksi akseli kiinnitettynä runkoon, kun taas karteesisella robotilla on yksi tai kaksi lineaarista servoakselia kiinnitettynä runkoon, riippuen mallista. SCARA-robotin liikkeet ovat nopeampia verrattuna karteesisiin robotteihin. SCARA-robotin akselit mahdollistavat robotin taipumista pieneen tilaan ja ahtaisiin paikkoihin. Tästä on hyötyä suurissa robottiyksiköissä, joissa SCARA-robotti pystyy tekemään toisille roboteille tilaa ja näin ollen välttämään mahdolliset törmäykset.

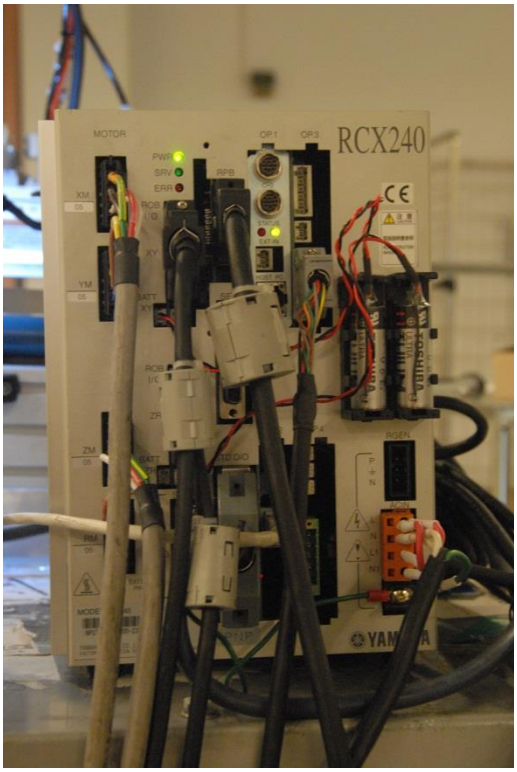
Kuva 6. Karteesinen robotti.

SCARA-robotin neljä vapausastetta muodostuvat kahdesta kääntyvästä akselista, yhdestä pystysuuntaisesta akselista ja pystysuuntaisen akselin kääntämisestä. Kääntyvistä akseleista johtuen robottia ohjataan joko vasen- tai oikeakätisenä. Ohjelman kiertoaikaan vaikuttaa hyvin paljon se, tarvitseeko syklissä vaihtaa robotin kätisyyttä lennossa. Tällöin robotti joutuu tekemään suuren liikkeen, joka ei ole suotavaa ohjelman kiertoaikaa silmällä pitäen.

2.4 Ohjain

Servo-vahvistin saa ohjeensa servo-ohjaimelta, joka tässä tapauksessa ohjaa SCARA-robotin kaikkia akseleita yhtä aikaa. Tällä saadaan aikaan mahdollisimman nopea ja tarkka paikoitusajo. SCARA-robotti voitaisiin myös rakentaa normaalille liikkeen ohjaimelle, jossa ajettaisiin neljän akselin lineaarisella interpolaatio-ajolla akseleita samanaikaisesti. Tällöin ohjelmassa pitäisi laskea kinematiikka, jolla muutettaisiin XYZR-koordinaattikäsky servoakseleille paikkakäskyiksi. Tätä paikkakäskyä muutettaisiin joka ohjelmakierroksella.

Servo-ohjainta (kuva 7) on mahdollista laajentaa erilaisilla lisäkorteilla kuten kenttäväylä-korteilla, konenäkölaajennuksella tai paikka-anturitulolla, jolla voidaan hallita kuljettimella liikkuvien kohteiden paikkatietoa.



Kuva 7.Ohjainyksikkö, johon on liitetty DeviceNet, seuranta ja Ethernet-laajennuskortit.

Työssä käytetyssä SCARA-robotin ohjainyksikössä oli liitettynä Ethernet- ja paikoituslisäkortit. Ethernet-kortilla hallittiin ulkopuolisen kamerajärjestelmän lähettämä tietoa, jossa tuli kappaleen paikka- ja suuntatiedot. Paikoituskortilla tiedetään puolestaan koko ajan, missä kyseinen kappale on, kun sen paikka on kerran ilmoitettu taulukkoon. Ethernet-laajennuskortilla olisi pystytty myös tekemään liittyminen muuhun robottisoluun, mutta tässä työssä käytettiin vain yhtä robottia ja konenäköjärjestelmää, jolloin ei tarvinnut tehdä liittymistä toiseen robottiin tai ohjelmoitavaan logiikkaan.

3 Konenäkö

Konenäöllä tutkitaan erilaisia kohteita kuvan analysointiin perustuen. Kohteesta otetaan kuva, jonka jälkeen etsitään tarpeelliset tiedot erilaisilla matemaattisilla menetelmillä. Tämän jälkeen lähetetään tarpeelliset tiedot eteenpäin käytettäväksi. Nykypäivänä voidaan tutkia monimutkaisempia asioita kuten usean kohteen samaan aikaan tunnistamista kohteesta, johtuen prosessorien laskentatehon kasvamisesta ja ohjelmien tunnistusmenetelmien tehokkuuden kasvamisesta.

Konenäköjärjestelmät muodostuvat valaistuksesta, kamerasta, objektiivista, prosessoriyksiköstä, jossa on konenäön sovellus sekä, kommunikointirajapinnasta. Konenäön hyvyys on jokaisen osa-alueen tulo [7].

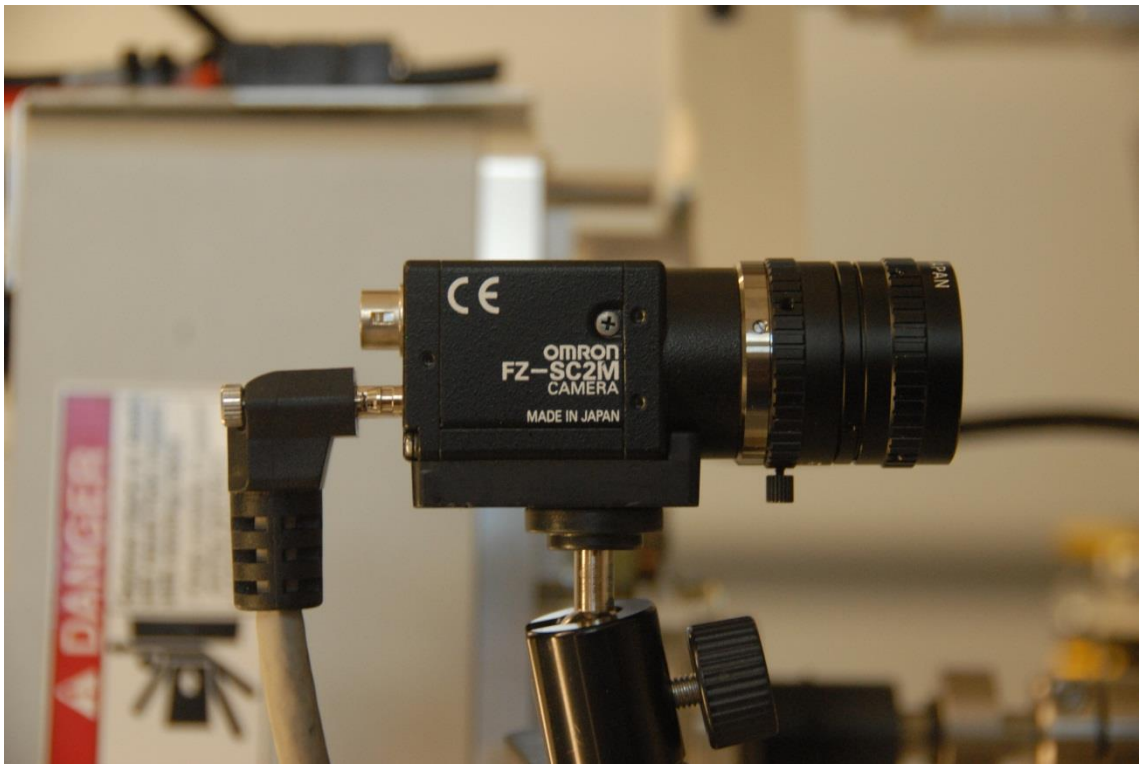
3.1 Valaistus

Valaistuksen tarkoituksena on saada kohdekuva tasalaatuiseksi ja erottamaan tarvittavat ominaisuudet taustasta. Tässä työssä käytin suoraa diffuusiovalaistusta, jolla saavutettiin kappaleen hyvä erottuminen taustasta ilman varjoja, jotka olisivat vääristäneet mahdollisesti tulosta. Valaistusta valittaessa on myös hyvä huomioida, ettei valo syki verkkotaajuuden mukana, kuten loisteputket vilkkuvat 50 Hz taajuudella.

Diffuusiovalaistuksen ideana on saada kohteeseen tasainen valaistus kappaleen jokaisesta suunnasta, jolloin ei muodostu varjoja kappaleen reunoille. Huonona puolena tässä on kontrastin huonominen taustaan verrattuna.

3.2 Kamera

Kameroita on saatavina erityisen nopeista kameroista värikuvallisiin ja suuriresoluutioisiin kameroihin. Tässä työssä ei ollut erityistä tarvetta erikoisnopealle kameralle, joten valitsin kameraksi värikameran 2 miljoonan pikselin CCD-kuvaelementillä (kuva 8).



Kuva 8. Omronin FZ-SC2M-värikamera.

Kameran käytettävät pikselit ovat 1600x1200. Kuvan fyysiseen kokoon vaikuttaa objektiivi, joka tulee valita kohteen mukaan sopivaksi. Kameran valintaan vaikuttivat kappaleet, jotka ovat keltaisia ja joiden tausta on vihreä, jolloin värikameralla saadaan kohteet erottumaan toisistaan helposti ja luotettavasti.

CCD-kuvaelementti muodostuu tässä kamerassa 1600x1200 puolijohdeanturista. Nämä puolijohdeanturit muodostavat kuvan, joka siirretään kuvan prosessointiohjelmalle prosessoitavaksi siirtorekisterin kautta.

3.3 Suodattimet

Kuvan suodattimia on kahden tyyppisiä: fyysisiä ja ohjelmallisia. Fyysisillä suodattimilla valokuvauksessa saadaan esimerkiksi korostettua taivaan sinisyyttä. Tässä työssä ei tarvinnut käyttää fyysisiä suodattimia, koska kohde ei ollut kiiltävä kappale ja se erottuu tarpeeksi hyvin taustasta. Sen sijaan ohjelmallisilla suodattimilla parannettiin kuvaa, jolloin kappale erottui taustasta erinomaisesti.

Ohjelmallisilla suodattimilla saatiin tausta suodatettua kokonaan mustaksi ja kappale valkoiseksi, jolloin kuva oli binaarisoitua. Tällöin kuvan pikseleillä on vain kaksi arvoa, joko

0 tai 1. Binäärisoidun kuvan käsittely ohjelmassa on huomattavasti nopeampaa kuin 24-bittisen värikuvan käsittely.

4 Ethernet-liikenne

Ethernet-liikenne on kasvanut nykypäivän teollisuudessa ja arkipäiväisessä tiedonsiirrossa merkittäväksi tekijäksi perinteisten sarjaliikennekommunikoiden rinnalle. Ethernet liikenteeseen on kasvanut kaksi ydintekniikkaa: TCP/IP- ja UDP-kommunikoinnit. TCP-kommunikointi muodostuu sanoista Transmission Control Protocol ja Internet Protocol. TCP/IP-kommunikointia käytetään nykypäivänä kohteissa, joissa tarvitaan luotettavaa tiedonsiirtoa. Mikäli luotettava tiedonsiirto ei ole pakollinen, voidaan käyttää UDP (User Datagram Protocol) -kommunikointia. [6]

4.1 IP-osoite

IP-osoite on oleellinen osa ethernet-kommunikointia. IP-osoite on laitteen osoite, johon viitataan, kun halutaan liikennöidä kyseisen laitteen kanssa. Samassa verkossa ei voi olla kuin yksi laite tietyllä IP-osoitteella. IP-osoite muodostuu kahdesta osasta: verkon ja laitteen numerosta. IP-osoiteilla on 32bittiä IPv4-standardissa. IP-osoitteet on jaettu viiteen eri luokkaan: A,B,C,D ja E. A-luokan mukaisessa soitteessa osoitteen seitsemän ensimmäistä bittiä kuvaavat verkkoa ja loput bitit laitteen osoitetta, jolloin on mahdollista saavuttaa yli 2 miljardia eri osoitetta. B-luokan osoitteissa puolestaan 14 ensimmäistä bittiä on varattu verkolle ja loput on laitteille käytettävissä, jolloin saadaan yli miljardi osoitetta. C-luokan osoitteissa käytetään 25-bittiä verkon määrittelyyn ja loput laitteille, jolloin saadaan yli puoli miljardia osoitetta. D-luokan osoitteet on varattu ryhmälähetyksiä varten ja E-luokan osoitteet on varattu tulevaisuutta varten. [6]

Varattuja IP-osoitteita ovat osoitteet, joissa kaikki bitit ovat nollia. Tällä tarkoitetaan tämänhetkistä verkkoa. Tämä IP-osoite on varattu palvelimille, jotka lähettävät dataa verkkoon, mutta eivät tiedä vielä kohdetta. Osoite, jossa laitteen osiossa kaikki bitit ovat ykkösiä, on varattu tiedon siirtoon, jossa on tarkoitus lähettää kaikille palvelimille tietoa. Osoite 127.0.0.0 tarkoittaa omaa konetta. [6]

IP-osoitteen otsikko-osassa määritellään versio, kuinka pitkä otsikko on, viestin tärkeys, jotta viesti menee tarpeeksi nopeasti verkon läpi, viestin pituus, tunnistenumero, onko

viesti pilkottu useampaan osaan, viestin elossa pitoaika, protokollatunnus, tarkistusnumero sekä tietenkin lähettäjän IP-osoite ja vastaanottajan IP-osoitteet.

4.2 QOS-palvelu

Viestin tärkeys on olennainen osa tiedonsiirrossa. Tällä määritellään ruuhkaisessa palvelussa, kuinka nopeasti viesti menee läpi verkosta. Mikäli viestillä kestää kauan kulkea verkossa, saattaa viestin tieto olla jo vanhentunut, ennen kuin se pääsee perille käsiteltäväksi. Tästä syystä verkossa kulkevassa tiedossa on määritelty sen tärkeys, jolloin reitittimet osaavat käsitellä tiedon tärkeysjärjestyksessä. [6]

IP-osoitteen otsikossa on oma kohta tähän tarkoitukseen. Standardin mukana on 8 eri tasoa, joista tärkeimpänä ovat verkon hallintaan kuuluvat viestit ja pienimpänä rutiiniviestit. Standardissa on sanottu seuraavasti:
Tärkeysjärjestys pienimmästä suurimpaan:

- 000 Rutiini
- 001 Tärkeä
- 010 Välitön
- 011 Salamana
- 100 Salamana yliajo
- 101 Kriittinen
- 110 Internetverkon hallinta
- 111 Verkon hallinta

4.3 TCP/IP-kommunikointi

TCP/IP-kommunikointi muodostuu sanoista Transmission Control Protocol sekä Internet Protocol, niin kuin aikaisemminkin mainittiin. TCP/IP on yhteysorientoinut kommunikointitapa eli yhteys pidetään auki, vaikka ei olisi mitään tietoa siirrettävänä laitteiden välillä.

TCP/IP:n lähettäessä tietoa toiselle ohjelmalle vastaanottajan tulee vastata positiivisella vastauksella tietyn aikarajan sisällä. Viesteille määritellään aika, jonka se on voimassa. TCP-kommunikoinnissa aikaraja lasketaan keskiarvosta, jolla kommunikointi liikennöi. Keskiarvon laskemin on tärkeä osa protokollaa. Tällöin saadaan luotettavampi kommunikointi riippumatta verkon kuormituksesta. TCP/IP-viestissä on nähtävissä tämä aika,

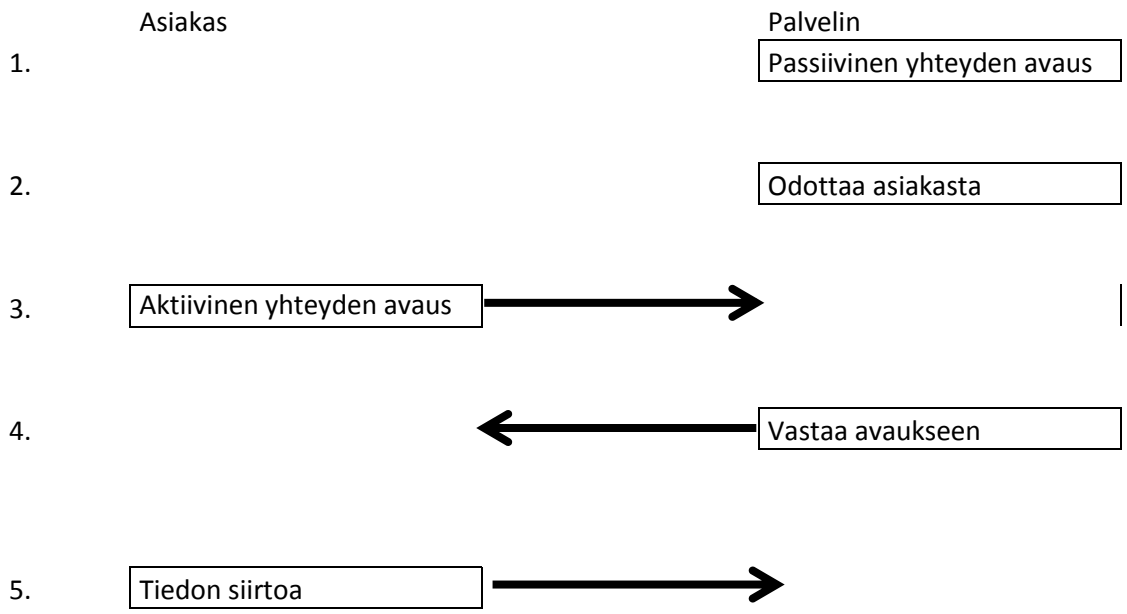
jolloin viestin on voimassa (kuva 9). Mikäli vastausta ei saada TCP-protokollan mukaan, lähetetään uudestaan viesti, johon ei ole saatu positiivista vastausta. Vastauksessa vastaanottaja kertoo myös, kuinka paljon dataa se voi ottaa vastaan, ettei tiedon käsittelyssä tule ongelmia kuten viesti puskurin ylivuotona. [6]

```
⊞ Frame 12: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
⊞ Ethernet II, Src: Pfu_4d:06:ca (00:80:17:4d:06:ca), Dst: YamahaMo_01:01:29 (00:04:c6:01:01:29)
⊟ Internet Protocol Version 4, Src: 192.168.0.100 (192.168.0.100), Dst: 192.168.0.254 (192.168.0.254)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 66
    Identification: 0x8151 (33105)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊞ Header checksum: 0xf6b1 [correct]
    Source: 192.168.0.100 (192.168.0.100)
    Destination: 192.168.0.254 (192.168.0.254)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ Transmission Control Protocol, Src Port: netinfo-local (1033), Dst Port: telnet (23), Seq: 34, Ack: 7, Len: 26
⊞ Telnet

0000  00 04 c6 01 01 29 00 80  17 4d 06 ca 08 00 45 00   .....).. .M....E.
0010  00 42 81 51 40 00 80 06  f6 b1 c0 a8 00 64 c0 a8   .B.Q@... .....d..
0020  00 fe 04 09 00 17 8c 97  3a a3 00 0c 97 06 50 18   ........ :.....P.
0030  3f 92 c6 a8 00 00 50 31  3b 20 3d 39 34 38 2e 35   ?.....P1 ; =948.5
0040  38 2c 35 33 36 2e 32 30  2c 2d 31 31 2e 36 38 0d   8,536.20 ,-11.68.
```

Kuva 9. TCP/IP-kommunikoinnista kaapattu tiedonsiirto.

Ennen kuin tieto voidaan siirtää laitteiden välillä, tarvitsee muodostaa yhteys. Yhteyden muodostuksessa toinen laite toimii palvelimena ja toinen toimii asiakkaana. Kaksi asiakasta ei voi siirtää tietoa keskenään, jollei laitteiden välillä ole palvelinta. TCP/IP-protokollassa ei ole teollisuusväylistä tuttua isäntä/orja-asetelmaa, jossa isäntä kysyy ja orja vastaa, vaan kummatkin voivat lähettää tietoa. [6]



Kuva 10. TCP/IP-yhteyden avaus

Yhteyden muodostuksessa palvelimen tulee avata passiivinen kommunikointiportti. Passiivisen yhteyden avauksessa tarvitsee tietää IP-osoite, paikallinen kommunikointiportti ja mahdollinen aikaraja kommunikoinnille. Tämän jälkeen palvelin jää odottamaan asiakkaalta yhteyden avausta eli seuraavaksi asiakas avaa aktiivisen yhteyden. Tähän palvelin vastaa asiakkaalle. Tämän jälkeen on mahdollista siirtää tietoa asiakkaan ja serverin välillä (kuva 10). [6]

4.4 UDP-kommunikointi

UDP tarjoaa TCP/IP-protokollaan verrattuna todella vähän, mikä tekee siitä nopeamman. UDP ei tarjoa tarkistusmenetelmiä viestin eheyden selvittämiseen, eikä virhetilanteista selviämiseen toisin kuin TCP/IP, joka tarkistaa vastauksen ja lähettää uudestaan tarvittaessa. Tästä johtuen ohjelmassa joudutaan suorittamaan tarkistuslaskelmat, jolla katsotaan, että viesti meni perille ehjänä. UPD-kommunikoinnissa määritellään portti, josta ohjelma kommunikoi toisen ohjelman kanssa. Tästä johtuen UDP-otsikko-osassa määritellään lähetetyn portin numero, vastaanottavan porttinumero ja pituus. Lisäksi voidaan määritellä tarkistuskohta, jolloin voidaan verrata, tuliko tieto ehjänä perille. [6]

5 Robotin ohjelma

Työssä käytetty robotti oli ensimmäinen Suomeen saapunut Yamahan SCARA-robotti. Tästä johtuen ensimmäisinä tehtävinä oli saada se liikkeelle ja ymmärrettyä, miten kyseinen robotti ohjelmoidaan, mitkä ovat robotin ominaisuudet sekä miten ohjelmointiohjelmaa käytetään.

Ensimmäinen robotille tehty ohjelma oli CF-muistikorttien liikuttelua paikasta toiseen. Tällöin pystyttiin toteamaan robotin nopeus ja tarkkuus valmistajan ilmoitusten mukaiseksi. Tästä saatiin myös hyvä demo-ohjelma asiakkaille näytettäväksi. Samalla se toimi hyvänä harjoitustyönä tulevaa demoa varten.

Robotin ohjelmat tehtiin aluksi pienissä osissa, jolloin kokonaisuuden hallitseminen oli helpompaa. Pieniksi ohjelmiksi muodostui kommunikointi konenäön ja robotin välille, kappaleen seuraaminen kuljettimella ja kappaleiden pinoaminen toiselle kuljettimelle. Lopuksi yhdistettiin nämä kaikki yhteen ohjelmaan ja korjattiin virheet, joita tuli tehtyä, kun yhdistettiin ohjelmat yhdeksi kokonaisuudeksi.

5.1 Konenäön ja robotin välinen kommunikointi

Robotin ja konenäön välinen kommunikointi hoitui TCP/IP:tä käyttäen. Robotti toimi TCP/IP-asiakkaana ja konenäkö toimi palvelimena. Testialustassa ei ollut mahdollista kytkeä erillistä liipaisutuloa konenäköjärjestelmälle, joten liipaisu tuli suorittaa aikapohjaisena. Tästä johtuen robotin ohjelmaan joutui tekemään suodattimen, ettei samaa kappaletta rekisteröity useampaan otteeseen muistipuskuriin. [12]

Robotin ohjelmassa käytettiin käskyä SEND <muuttuja> TO ETH, jolla lähetetään tietoa vastaanottajalle. ETH tässä tapauksessa tarkoittaa ennalta määriteltyä kohdetta, joka on konenäköjärjestelmä. SEND ETH to RCV$ tarkoittaa, että luetaan portin puskurista tiedot RCV$-muistipaikkaan ja siitä pilkotaan tiedot tarpeellisiin osiin ja lähetetään ne ohjelmalle käsiteltäväksi.[8]

Ohjelmassa lähetetään SEND ”M” to ETH, joka on konenäölle kuvanottokäsky, jolloin kamera ottaa kuvan ja lähettää sen takaisin. Tämä vastaus otetaan SEND ETH TO RCV$, jolloin vastaus on purettavissa pienempiin osiin RCV$-muuttujasta. [8]

Konenäön ja SACARA-robotin välisessä kommunikoinnissa ilmeni ongelma, jonka alkuperää ei saatu selville. SCARA-robotti ja konenäkö jäivät joskus lähettämään toisilleen virhettä, joka kuormitti verkkoa, mutta ei vaikuttanut robotin eikä konenäön toimintaan. SCARA saattoi joskus lähettää viestin: 14.22:No start code(@)\r\n (Kuva 11), johon taas konenäköjärjestelmä vastasi ER-viestillä, koska se ei tunnista robotin virhekoodia omaksi käskykseen. Siihen taas robotti vastasi samalla viestillä. Tästä syntyi ikuinen kierre, joka kuormitti turhaan reititintä.

```
29 2.966346   192.168.0.254   192.168.0.100   TELNET   78 Telnet Data ...

⊞ Frame 29: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊞ Ethernet II, Src: YamahaMo_01:01:29 (00:04:c6:01:01:29), Dst: Pfu_4d:06:ca (00:80:17:4d:06:ca)
⊟ Internet Protocol Version 4, Src: 192.168.0.254 (192.168.0.254), Dst: 192.168.0.100 (192.168.0.100)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 64
    Identification: 0xe4e2 (58594)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 32
    Protocol: TCP (6)
  ⊞ Header checksum: 0x3323 [correct]
    Source: 192.168.0.254 (192.168.0.254)
    Destination: 192.168.0.100 (192.168.0.100)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: netinfo-local (1033), Seq: 16, Ack: 147, Len: 24
⊟ Telnet
    Data: 14.22:No start code(@)\r\n

0000  00 80 17 4d 06 ca 00 04  c6 01 01 29 08 00 45 00   ...M.... ...)..E.
0010  00 40 e4 e2 00 00 20 06  33 23 c0 a8 00 fe c0 a8   .@.... . 3#......
0020  00 64 00 17 04 09 00 0c  97 0f 8c 97 3b 14 50 18   .d...... ....;.P.
0030  01 c0 9e de 00 00 31 34  2e 32 32 3a 4e 6f 20 73   ......14 .22:No s
0040  74 61 72 74 20 63 6f 64  65 28 40 29 0d 0a         tart cod e(@)..
```

Kuva 11. Virhekoodi verkkoliikenteessä.

5.2 Liikkuvan kohteen seuraaminen

Työn kannalta olennainen osa oli seurata kohdetta ja poimia se liikkeestä tarttujaan, jolloin saadaan pyöritettyä enemmän kappaleita ja tiheämpään tahtiin. Tällöin voidaan myös sijoittaa samalle linjalle useampi SCARA-robotti poimimaan kappaleita, jos niitä

tulee enemmän kuin yksi SCARA-robotti pystyy siirtämään. Tällaisia sovelluksia on teollisuudessa esimerkiksi konvehtirasioihin konvehtien laittaminen.

Ohjaimeen lisättiin R6YACTR01-lisäkortti, jolla pystytään liittämään inkrementti-paikka-anturi ohjaimeen. Tällöin ohjaimessa on komento CADDQUE <puskurin numero>, <pisteen tieto>, jolla lisätään kappale puskuriin, johon ohjain muuttaa kappaleen paikkaa automaattisesti, kun hihna pyörii. CADDQUE-komennossa on mahdollisuus lisätiedolle, jolloin voidaan määritellä alue, jonka sisällä ei saa olla toista kappaletta. Tällöin voidaan eliminoida päällekkäiset kappaleen rekisteröinnit muistirekisteriin. Ohjaimessa tarvitsee määritellä robotille, missä suunnassa kuljetin kulkee, jotta se osaa lisätä arvoa kappaleeseen oikeassa XYZ-suunnassa. [9;10]

CCHKQUE-käskyllä voidaan tarkkailla, onko kappale ennen työaluetta, työalueella, työalueen ohittanut vai onko se seurattavana. Tämän käskyn tulosta seuraamalla voidaan päätellä, mennäänkö kappaletta poimimaan vai poistetaanko se muistipuskurista. [10]

Ohjaimessa on käsky CTDRIVE (Korkeus jolla seurataan), jolla käsketään robottia seuraamaan kappaletta. Tällöin, kun robotti seuraa kappaletta, voidaan CTMOVE-käskyllä käskeä seurannan aikana muuttaa korkeutta, jolloin päästään tartuttavan kappaleen pinnan tasolle valmiina nostamaan. [10]

5.3 Pinoaminen

Pinoaminen voidaan toteuttaa tässä robotissa PMOVE-käskyllä, jolloin tarvitsee määrittää ohjelman alussa PDEF-komennolla, minkälainen on pinoamisalue. Lisäksi samalla määritellään, montako kappaletta tulee XYZ-suunnissa. Tässä työssä käytettiin vain aluetta 2x2x1, koska mekaaniset ominaisuudet rajoittivat työtä, joten PDEF(1)= 2,2,1. Lisäksi ennen PDEF-komentoa oli tarve määritellä Pallet-valikosta pisteet P1,P2,P3,P4 ja P5. Nämä pisteet määrittelevät kolmiulotteisen alueen, jonka sisään PDEF-komennolla lasketaan pisteet, joihin voidaan ajaa. [8;11]

Tämä voitaisiin tehdä myös matematiikalla, jolloin valittaisiin jokin piste robotin työalueelta, johon lisättäisiin x määrä matkaa joka kerta, kun tuotaisiin uusi kappale pinoamisalueelle. Esimerkissä on valittu arvot: x=100, y=100 ja z= 50. Ohjelma laskisi joka kerta, kun kappalenumero muuttuu edellisestä, uuden paikan lisäämällä x+50, ja mikäli kappalenumero on 2, lisätään seuraavalle kappaleelle (3) paikkaan myös y+50 ja laitettaisiin x arvo 50, jolloin kappaleen 3 paikka tulee olemaan x=100, y=150 ja z=50. [8]

```
IF kappalenumero <> Edellinenkappale THEN
x=x+50
        IF kappalenumero = 1 THEN
                x=100
                y=100
                z=50
        ENDIF
        LOCX(P5)=x
        LOCY(P5)=y
        LOCZ(P5)=z
        MOVE P5
        IF kappalenumero=2 THEN
                y=y+50
                x=50
        ENDIF
ENDIF
```

Kummassakin tavassa on omat hyvät puolensa. Ensimmäisessä, jossa määritellään pisteet, jotka muodostavat kolmiulotteisen pinoamisalueen, on helppo tehdä esimerkiksi erikokoisille tuotteille pinoamisohjelma, jolloin vain ohjelman alussa määritellään PDEF-komennolla samalle alueelle joko enemmän tai vähemmän kappaleita. Huonona puolena on tässä ohjelmoijan kannalta virheiden selvittäminen. Jälkimäisessä hyvänä puolena on se, että robotti tekee aivan varmasti sen, minkä ohjelmoija on kirjoittanut. Huonona puolena on sen monimutkaisuus, mikäli tarvitsee tehdä monimutkaisempia pinoamistehtäviä.

Tämän takia päädyttiin työssä käyttämään SCARA-robotissa olleita valmiita PDEF- ja PMOVE-käskyjä. Tällöin esiteltävä ohjelma olisi helppo muuttaa kappaleita vaihdattaessa toisiin.

Työssä ilmeni mekaaninen ongelma, joka aiheutti pinoamisessa 1-2 millimetrin heittoja. Ongelmana oli robotin liikkeistä aiheutuva kameran heiluminen. Tällöin kamera ja kuljetin eivät pysyneet samassa kohdassa, mikä ilmeni virheenä. Robotin alusta pääsi jousta-

maan nopeiden liikkeiden aikana, jolloin koko solu pääsi heilumaan. Tämäkin saattoi liikuttaa poimittavia kappaleita kuljettimella. Nämä seikat tulee ottaa huomioon suunnitellessa teollisuuteen robottisoluja.

6 Konenäön ohjelma

Konenäön ensimmäinen ohjelman kiertoaika oli noin 900 ms. Tämä aika on tuotantokoneelle aivan liian suuri. Pelkällä etsintätoiminolla sai testattua konenäköjärjestelmän ja robotin välisen kommunikoinnin toimivuutta, joten ohjelmaan ei aluksi panostettu. Myöhemmin muiden osa-alueiden ollessa kunnossa paneuduttiin ohjelmaan enemmän. Tällöin saatiin ohjelman kiertoaika 10 kertaa nopeammaksi, mikä on jo siedettävä tähän käyttötarkoitukseen.

Ohjelman kiertoaikaa pienentäessä suurimpina tekijöinä oli se, että kuvasta tehtiin binaarikuva. Tällöin kaikkien kappaleiden etsintätoiminon ei tarvinnut käsitellä niin paljoa tietoa. Toisena tekijänä ohjelman kiertoaikaa pudotettaessa oli position compensation –toiminnon käyttö, jolloin siirrettiin kappale aina keskelle ruutua käsiteltäväksi. Tällöin etsintätoiminnon ei tarvitse etsiä kappaletta niin suurelta alueelta, jolloin taas tiedon käsittelyn määrä putosi.

Omronin FZ3-konenäköjärjestelmä oli aluksi TCP/IP-palvelin ja SCARA-robotti oli myös palvelin. Kaksi palvelinta ei voi keskustella keskenään ilman TCP/IP-asiakasta. Tässä olisi useampikin mahdollisuus ratkaista ongelma esimerkiksi tekemällä erillinen logiikka, joka kommunikoi kummankin laitteen välillä ja vaihtaa tarvittavat tiedot näiden välillä. Tästä olisi aiheutunut lisäkuluja pieniin soluihin, joten ongelman ratkaisuksi muodostui jommankumman laitteet muuttaminen TCP/IP-asiakkaaksi.

Konenäön muuttaminen TCP/IP-asiakkaaksi muodostui helpommaksi toteuttaa johtuen laitteiden kehittämispaikoista. Samalla kun konenäkö muutettiin TCP/IP-asiakkaaksi, lisättiin ohjelmaan STRING-muuttujan lähetyskäsky. Tällöin pystyttiin muodostamaan X- ja Y-koordinaateista STRING-muuttuja, joka lähetettiin robotille.

7 Loppusanat

Työssä saavutettiin asetetut tavoitteet. Tärkeimpänä tavoitteena oli saada SCARA-robotti toimimaan ilman ylimääräistä logiikkaa tai tietokoneohjelmaa konenäkö-järjestelmän kanssa. Toisena tärkeänä tavoitteena oli saada aikaiseksi ohjelma, jota voisi helposti muokata asiakkaiden tarpeiden mukaiseksi ja joka voisi toimia hyvänä SCARA-robotin esittelyohjelmana.

Mikään projekti ei ole täydellinen, joten tähänkin jäi paljon parannettavaa, jota ei ehditty tiukan aikataulun takia tekemään. SCARA-robotin ja konenäköjärjestelmän välisessä kommunikoinnissa ylimääräiset virheviestit jäivät korjattavaksi myöhemmäksi. Lisäksi kameran asennus olisi tehtävä paremmin. Kyseisessä esittelyohjelmassa kamera pääsi heilumaan, kun robotti liikkui. Tästä aiheutui noin 1-2 mm heittoa pinoamisalueella. Lisäksi tässä työssä robotti oli asennettuna metallisen laatikon päälle, joka pääsi heilumaan.

Vuonna 2013 tässä tuotettua materiaalia on käytetty asiakkaiden koulutukseen Suomessa ja Virossa. Konenäköjärjestelmään tehty TCP/IP-kommunikointi muutokset on sisällytetty normaaliin sulautettuun hallintaohjelmistoon, jolloin sama sulautettu hallintaohjelmisto hoitaa TCP/IP-client- ja TCP/IP-server-kommunikoinnit.

Lähteet

1 What is a Servomotor? Verkkodokumentti http://blog.fast-tobuy.com/2012/06/11/what-is-a-servomotor/#.UPgjQSdLPaY .Luettu 21.12.2012

2 Drury, Bill. 2009.The Control Techniques Drives and Controls Handbook. IET

3 Ruiz, Carles. 2012. Power stage in a generic servodrive. Omron sisäinen dokumentti

4 Incremental Endcoders. Verkkodokumentti http://www.pc-control.co.uk/incremental_encoders.htm Luettu 30.12.2012

5 New Encoder Track Design Improves Position Control. Verkkodokumentti http://www.designworldonline.com/new-encoder-track-design-improves-position-control/ Luettu 27.12.2012

6 Rodriguez, Adolfo. 2001. TCP/IP Tutorial and Technical Overview. IBM

7 Gonzalez, Rafael. 1993. Digital Image Processing. Addison Wesley

8 Omron, 2010. I139-EN-01+SCARA-YRC-Series+Program

9 Omron, 2010. I140-EN-01+SCARA-YRC-Series+UserManual

10 Omron, 2010. I147-EN-01+SCARA-Vision-Tracking+UserManual

11 Omron, 2010. I148-EN-01+SCARA-Studio+UserManual

12 Omron, 2010. I151E-EN-01+SCARA-Ethernet+UserManual

SCARA-robotin ohjlema

```
'---Asettelut----
SERVO ON
LEFTY        'ASETETAAN VASENKÄTISEKSI
'
CTCONTRL ON    'Tracking toiminto päälle
CTVISION OFF    'EI KÄYTETÄ OMAA VIDEO TULOA
'
SPEED 80
ACCEL 70
DECEL 70
'
PDEF(7) = 2,2,1
P20= 0.00 0.00 0.00 0.00 0.00 0.00
MOVE P,P1
'
'---MUUTTUJIEN ASETTELU---
A=1 'PINOAMISESSA KÄTETTÄVÄ APUBITTI
'
START *KAMERA,T2 'ALIOHJELMA 2
START *OUTOFRANGE,T3 'ALIOHJELMA 3
*KULJETIN:
IF CCHKQUE(1)>0 THEN
CTMOVE 1, Z=0.00
'
   IF CCHKQUE(1)=2 THEN
      CTDRIVE(4.55)'LASKEE ALAS
      CALL *IMU
      CTDRIVE(0.00)'OPTIMOI KORKEUS
      'CALL *PINOUS
      MOVE P,P1005,Z=0.00
      PRINT "PINOUS"
      PRINT "PINOUS NUMERO "; A
      'JOS RIVITYS ON TÄYNNÄ NOLLATAAN TÄYTTÖ
```

```
      IF A>4  OR A<0 THEN A=1
'
      PMOVE(7,A),Z=0
      A=A+1
    ENDIF
      CALL *PUHALLUS
      CRMVQUE 1
      MOVE P,P2,Z=0.00 'SIIRRETÄÄN STANDBY PAIKKAAN
'
ENDIF
GOTO *KULJETIN
'
*KAMERA:
'LÄHETETÄÄN KÄSKY
SEND "M" TO ETH
SEND ETH TO RCV$
PRINT RCV$
SEND ETH TO RCV$
PRINT RCV$
VX$=MID$(RCV$,6,6)
VY$=MID$(RCV$,13,6)
VR$=MID$(RCV$,21,6)
WX=VAL(VX$)
WY=VAL(VY$)
WR=VAL(VR$)
'
IF WX<0 AND WY<130 AND WY>50 THEN
   PRINT WX;" ";WY;" ";WR
   'SIJOITETAAN P20 PAIKKAAN X/Y/R
   LOCX(P20)=15+WX
   LOCY(P20)=WY
   LOCR(P20)=WR+180
   CADDQUE 1,P20
   DELAY 1700 'ei tarvita anturi trikkauksessa
ENDIF
```

```
DELAY 300
GOTO *KAMERA
'
*OUTOFRANGE:
'---ULOSMENNEIDEN KAPPALEIDEN poisto
WHILE CCHKQUE(1)<0
CRMVQUE 1
PRINT "YLITE"
WEND
GOTO *OUTOFRANGE
'
'---ALIOHJELMAT---
SUB *IMU 'IMUPÄÄLLE
DO(20)=0
DO(21)=1
DELAY 50
END SUB
'
SUB *PUHALLUS 'PUHALTAA KAPPALEEN IRTI
DELAY 150
DO(20)=1
DO(21)=0
DELAY 50
END SUB
```